Physics-Aware Informative Coverage Planning for Autonomous Vehicles

Michael J. Kuhlman[1], *Student Member, IEEE*, Petr Švec[2], *Member, IEEE*, Krishnanand N. Kaipa[2], *Member, IEEE*, Donald Sofge[3], *Member, IEEE*, and Satyandra K. Gupta[4], *Senior Member, IEEE*

***Abstract*— Unmanned vehicles are emerging as an attractive tool for persistent monitoring tasks of a given area, but need automated planning capabilities for effective unattended deployment. Such an automated planner needs to generate collision-free coverage paths by steering waypoints to locations that both minimize the path length and maximize the amount of information gathered along the path. The approach presented in this paper significantly extends prior work and handles motion uncertainty of an unmanned vehicle and the presence of obstacles by using a Markov Decision Process based approach to generate collision-free paths. Simulation results show that the proposed approach is robust to significant motion uncertainties and reduces the probability of collision with obstacles in the environment.**

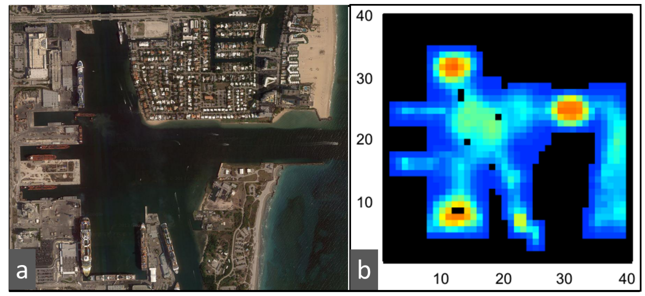


Fig. 1. (a) An example of a harbor patrol environment with multiple entry points for intruders (A harbor in Hollywood, FL; source: Map data ©2013 Google). (b) Obstacle regions (black) and the "information value" map

I. INTRODUCTION

There are many applications that need persistent monitoring of a given area, requiring repeated travel over the area to gather new information. Unmanned vehicles are well-suited for performing such tasks, but require automated planning. Moreover, certain locations in the area are designated as key locations and hence are more valuable from the information gathering perspective. Surveillance platforms in persistent monitoring tasks vary from aerial vehicles to surface vehicles (e.g., boats) depending upon the application.

Consider the use of an unmanned surface vehicle (USV) conducting harbor patrols to detect intruders. It is reasonable to assume that possible intruders will enter the harbor (Fig. 1(a)) from certain locations such as harbor entrances and shipping channels. This suggests the use of an "information value map" (see Fig. 1(b)) that signifies how some regions are more dynamic or interesting and should be observed more often. Though the underlying technical approach developed in this paper is intended for application to USV harbor patrolling, it is applicable to many different domains.

The problem of traveling over an area and gathering information can be viewed as a coverage planning problem [1]. There are many practical and theoretical challenges in solving the coverage planning problem in the context of regions of varying information value. It is possible for static obstacles to exist around the harbor, such as shoals. Further, windy conditions or swift currents can contribute significant uncertainty to the USV's location and motion, compounding the problem. This suggests using a physics-aware planner that is capable of planning under motion uncertainty while avoiding obstacles in the environment.

We begin with a short review of existing techniques for informative path planning for coverage planning problems. Branch and bound search was used in [2] to maximize the "informativeness" of a plan. In [3], Fisher information matrices, combined with rapidly-exploring random trees (RRTs), was used for information-rich path planning.

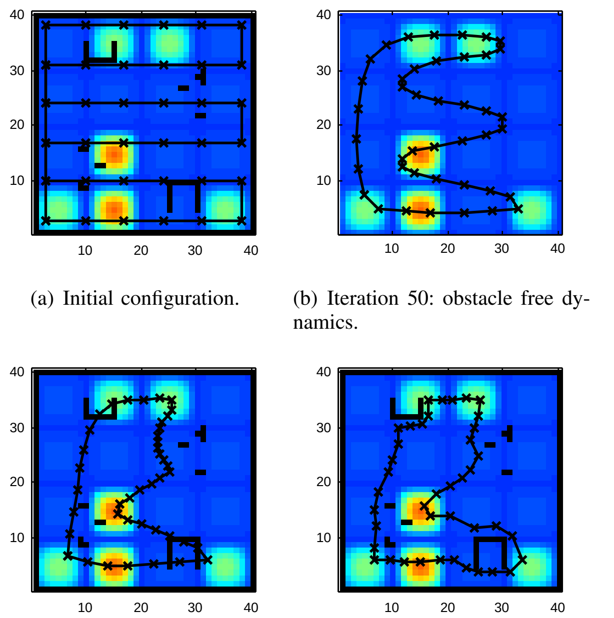
Persistent sensing approaches [4] model the information uncertainty in the environment as a field defined over a set of locations and assume that the field increases linearly at locations beyond the sensing range of the robot and decreases linearly at locations within the robot's range. Control-based approaches involving locational optimization include [5]–[7]. There also exist information-theoretic approaches that specify which areas are of interest and attempt to maximize the informativeness of a plan. In [8], agents followed the gradient of mutual information to minimize total entropy. In [9], a data structure called a probabilistic quad tree was proposed and used to provide a tree structure to the total entropy in the environment during target search. Other examples of information-theoretic coverage planning work include [10], [11]. Another approach to solve the problem of interest is direct planning using graph search. Conceptually, one could pick areas of interest in the environment and find the optimal path connecting fixed nodes, which is equivalent to solving the Traveling Salesman Problem (TSP). While TSP is an NP-complete problem, many approximate solutions exist and

[1]M.J. Kuhlman is with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742, USA and the Laboratory for Autonomous Systems Research, Naval Research Laboratory, Washington, DC 20375, USA mkuhlman@umd.edu

[2]P. Švec and K.N. Kaipa are with the Department of Mechanical Engineering, University of Maryland, College Park, MD 20742, USA {petrsvec,kkrishna}@umd.edu

[3]D. Sofge is with Navy Center for Applied Research in Artificial Intelligence, Naval Research Laboratory, Washington, DC 20375, USA donald.sofge@nrl.navy.mil

[4]S.K. Gupta is with the Department of Mechanical Engineering and Institute for Systems Research, University of Maryland, College Park, MD 20742, USA skgupta@umd.edu



(a) Initial configuration.

(b) Iteration 50: obstacle free dynamics.

(c) Iteration 431: end of obstacle free dynamics.

(d) Proposed modification.

Fig. 2. An example simulation of the proposed algorithm for generating collision-free informative paths under motion uncertainty. The background colors indicate the sensor function, connected black "x's" the waypoints, and the black cells are (impassable) obstacles.

have been used in path planning [12].

After analyzing direct planning approaches, we noticed several challenges associated with them. The nodes need not necessarily be fixed in place, which is a common assumption for search problems; new information coming from sensors could update the node locations as areas become no longer interesting while persistently monitoring an area. Alternatively, the traversal cost to reach a node could be discovered to be prohibitive, yet a nearby location could be reached to gather sufficient information while maintaining lower path traversal cost. All these factors point to challenges in selecting fixed locations to visit without considering the path length/information gain trade-off.

We believe that a more continuous, online update policy that also allows the locations of nodes to change in Euclidean space is preferable in our application. In fact, this approach was used in the work by Soltero *et al.* [5] and our work in this paper is inspired by their approach. We extend their approach by developing a waypoint feedback policy that morphs the informative coverage plan to: (1) minimize *expected path traversal cost*, accounting for motion uncertainty and presence of obstacles, and (2) maximize information gathered along the path. We formulate the problem using a Markov Decision Process (MDP) based approach (See Section II-A, [13], [14]). Paths generated using this approach can be executed on autonomous unmanned surface vehicles described in our prior work [15].

II. Problem Formulation & Approach

In this section we describe an approach for computing an informative coverage path for a vehicle operating in an environment with static obstacles and motion uncertainty. Let $X = \{(x, y)\} \subset \mathbb{R}^2$ be the continuous state space. Let $X_d = \{(x_i, y_i)\} \subset X$ be the finite, discrete regular grid. Let $\mathcal{E} = (\phi, \mathcal{O})$ be the environment, where $\phi : X \to \mathbb{R}^+$ is the sensor function that assigns the density of information in each state, and $\mathcal{O} \subset X_d$ is the set of states occupied by obstacles. Let $\tau = \{p_i\}_{i=1}^N \subset X$ be the informative coverage path that consists of a sequence of N waypoints. Finally, since this is a persistent monitoring task, we assume that τ is a closed path, where $i = N + 1$ is identified with $i = 1$.

The task is to update τ by steering waypoints to locations that both minimize path costs (e.g., sum of path lengths between neighboring waypoints) and maximize the amount of information gathered along the path. These competing objectives are formulated as a single-objective optimization problem in Section II-B and is similar to the approach proposed in [5]. However, we extend the previously used notion of minimizing path length to factor motion uncertainty and the presence of obstacles by minimizing *expected path traversal cost*. Generating a single path is not sufficient to account for motion uncertainty. Since the vehicle can deviate from a planned path during execution, use of a *feedback plan* which consists of a collection of state action pairs (x_i, u_i) is needed for each $x_i \in X_d$.

We have formulated a Markov Decision Process (MDP) problem for each waypoint so the vehicle generates collision free feedback plans to traverse from waypoint to waypoint and can predict expected path traversal costs (see Section II-A). The MDP problem is solved using the probabilistic value iteration [13] and its solution (value function and feedback plan) is *directly integrated* into the coverage plan using the proposed algorithm outlined in Section II-C. Analysis in Section IV shows that this modification to the waypoint steering algorithm reduces the probability of collision while executing the coverage plan compared to previous informative coverage planning algorithms.

A. Markov Decision Process Formulation

A Markov Decision Process (MDP) is defined as the following tuple MDP $= (X_d, U_d, \Theta, L, X_{goal})$ [13]. The state space of the MDP is the regular grid $X_d \subset \mathbb{R}^2$, where $X_{goal} \subset X_d$ is a collection of goal states. Let the vehicle's action space be $U_d = \{-1, 0, 1\} \times \{-1, 0, 1\}$, which are the free choices the vehicle can make when moving in X_d. However, we assume that there is uncertainty in the vehicle's state transitions. Given time index k, nature's actions on the system state θ_k are defined such that the vehicle's state transitions satisfy $x_{k+1} = x_k + \theta_k$. Here, θ_k is a random variable that is drawn from the sample space $\Theta_k = \Theta(x_k, u_k)$ (see Fig. 3). The probabilities of nature's actions on the system state are defined as $\Pr[\theta_k | x_k, u_k]$. $L = L(x, u, \theta, \mathcal{O})$ is the cost functional for transitioning between states, which also encodes the position of obstacles

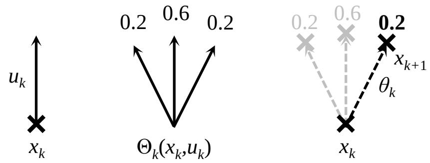


Fig. 3. A graphical depiction of possible state transitions from x_k to x_{k+1} conditioned on the control action u_k. Each $\theta_k \in \Theta(x_k, u_k)$ has an assigned probability. Note that u and θ need not reside within the same space.

by assigning infinite cost to movements that cause the vehicle to collide with obstacles.

Probabilistic Backwards Value Iteration (PBVI) is used to solve the MDP to generate the value function (expected path traversal cost-to-go) $\Phi : X_d \to \mathbb{R}^+$ and optimal feedback plan $\pi : X_d \to U_d$ [13]. In short, denote PBVI : MDP $\mapsto (\Phi, \pi)$, where Φ satisfies the Bellman equation (1).

$$\Phi(x) = \min_{u \in U} \mathbb{E}_\theta[L(x, u, \theta, \mathcal{O}) + \Phi(x + \theta(x, u))] \quad (1)$$

In (1), $\Phi(x)$ is minimized if the feedback plan $u = \pi(x)$ is executed. Assuming L is positive definite and deterministic (independent of nature's action, i.e., $L = L(x, \pi(x), \mathcal{O}) > 0$), it holds that $\Phi(x_k) = L(x_k, \pi(x_k)) + \mathbb{E}_{\theta_k}[\Phi(x_k + \theta_k)]$, or more importantly

$$\mathbb{E}_{\theta_k}[\Phi(x_{k+1})] < \Phi(x_k) \quad (2)$$

B. Objective Function and Waypoint Location Optimization

Let $\mathcal{H}$ be the single objective function defined in (3). The goal is to compute a waypoint-steering feedback policy $\dot{p}_i = u_i$ that minimizes $\mathcal{H}$.

$$\mathcal{H} = \sum_{i=1}^{n} \int_{V_i} \frac{W_s}{2} ||\mathbf{q} - p_i||^2 \phi(\mathbf{q}) d\mathbf{q} + \sum_{i=1}^{n} \frac{W_g}{2} [(\Phi_{p_{i-1}}(p_i))^2 + (\Phi_{p_{i+1}}(p_i))^2] \quad (3)$$

Here, W_s and W_g reflect the tradeoff between the competing objectives of steering waypoints to informative regions vs. reducing path length. Further, V_i is the Voronoi partition of waypoint p_i. Voronoi partitions have been frequently used in similar locational optimization problems [6], [16], [17]. (3) is similar to the Lyapunov-like function candidate found in [5], excluding the adaptation parameter.

The best physical interpretation of $\mathcal{H}$ is that it defines a spring-like potential energy so that virtual springs connect neighboring waypoints. Similarly, virtual springs connect waypoints to the informative center of mass of its respective Voronoi partition. Standard gradient descent technique cannot be used to locally minimize (3) due to the construction of Φ. Hence, we propose using the following continuous feedback policy (4):

$$\dot{p}_i = u_i = \frac{K_i}{\beta_i} [W_s M_{V_i} e_i + W_g (\Phi_{p_{i-1}}(p_i)) \hat{h}_{p_{i-1}}(p_i) + W_g (\Phi_{p_{i+1}}(p_i)) \hat{h}_{p_{i+1}}(p_i)] \quad (4)$$

Here, $M_{V_i} = \int_{V_i} W_s \phi(\mathbf{q}) d\mathbf{q}$ is the mass of the Voronoi partition V_i using the sensor function ϕ as the density function. $e_i = C_{V_i} - p_i$ is the vector difference of the Voronoi partition centroid and the waypoint p_i. $C_{V_i} = \frac{L_{V_i}}{M_{V_i}}$, where $L_{V_i} = \int_{V_i} W_s \mathbf{q} \phi(\mathbf{q}) d\mathbf{q}$ is the first moment. The integrals defined over the Voronoi partitions (e.g., M_{V_i} and C_{V_i}) are well approximated by the Riemann sum of points on the regular grid X_d. The Voronoi partitions were therefore approximated by calculating the nearest waypoint (i.e., the nearest neighbor) of each $x \in X_d$ using the k-nearest-neighbor (KNN) algorithm. K_i is a positive gain constant.

Also note that $\beta_i = M_{V_i} + 2W_g$ is a normalization parameter, and $\hat{h}_{p_j}(p_i)$ is the descent direction of Φ_{p_j}. (4) has the property that if $\hat{h}_{p_j}(p_i) = \nabla \Phi_{p_j}(p_i)$ then (4) is equivalent to normalized gradient descent of $\mathcal{H}$ (3).

When computing $\frac{\partial}{\partial p_i}(\Phi_{p_j}(p_i))^2$ in (4), it is assumed that the subscript p_j is fixed and hence $\frac{\partial}{\partial p_i}(\Phi_{p_j}(p_i))^2 = (2\Phi_{p_j}(p_i)) \nabla \Phi_{p_j}(p_i)$. Also, we assume $\frac{\partial}{\partial p_j}(\Phi_{p_j}(p_i)) = 0$ which explains the use of two terms in (3). The gradient $\nabla \Phi_{p_j}$ is not well defined since the domain of Φ is X_d. Instead of using gradient descent, we will define a descent direction $\hat{h}$ by averaging all possible outcomes θ when executing the feedback plan. It is reasonable to use either the mode (5) or expected value (6):

$$\hat{h}(x) := \hat{h}_{mode}(x) = \arg\max_{\theta \in \Theta} \Pr[\theta | x, \pi(x)] \quad (5)$$

$$\hat{h}(x) := \hat{h}_{mean}(x) = \mathbb{E}[\theta | x, \pi(x)] \quad (6)$$

However, verifying the desirable identity $\Phi_{p_j}(p_i + \eta \hat{h}) \overset{?}{\leq} \Phi_{p_j}(p_i)$ with $\eta \in (0, 1]$ is a difficult task since Φ is not convex with obstacles present. For example, we cannot invoke Jensen's inequality to show that, for example, $\Phi(\mathbb{E}[x + \theta]) \overset{?}{\leq} \mathbb{E}[\Phi(x + \theta)] < \Phi(x)$. It is possible to construct (perhaps pathological) counter examples where this desired inequality fails to hold. When calculating the descent direction $\hat{h}$, bilinear spline interpolation is used.

C. Proposed Algorithm

A direct execution of the waypoint feedback policy in (4), denoted as ICPS_MDP algorithm, results in a rubber-band like contraction of the waypoints wrapping around obstacles, with "attractor springs" pulling waypoints to the informative regions. To avoid local minima, obstacles are ignored during the computation of an initial informative coverage path. The algorithm considers obstacles in later planning stages during which it repairs the portions of the path that are invalidated by the obstacles, and optimizes the path with respect to the environmental effects that cause uncertainty in robot's motion.

Algorithm 1 COMPUTEINFORMATIVEPATH$(X_d, \mathcal{O}, \phi)$

Require: A discrete grid X_d, an obstacle region $\mathcal{O} \subset X_d$, and a sensor function ϕ.
Ensure: An informative coverage path τ.

Run informative path shaping algorithm ICPS(X_d, ϕ) [5] to compute an informative coverage path τ that ignores obstacles $\mathcal{O}$.
Run FINDCOLLISIONFREEWAYPOINTS$(\tau, \mathcal{O})$ to find collision-free neighboring waypoints of the waypoints that are inside $\mathcal{O}$.
Run the PBVI algorithm [13] to compute (Φ_{p_i}, π_{p_i}) of each waypoint $p_i \in \tau$. Update the informative coverage path τ between p_i and its predecessor p_{i-1} and add additional waypoints along the mode path connecting p_i and p_{i-1}.
Run ELIMINATEWAYPOINTS(τ, ϕ) to remove redundant (too close to other waypoints) or uninteresting waypoints (cover a region of the environment where little information is present).
Run ICPS_MDP(τ, ϕ) algorithm to smooth the informative coverage path.
Run ELIMINATEWAYPOINTS(τ, ϕ) to remove redundant or uninteresting waypoints.
Run ICPS_MDP(τ, ϕ) algorithm to smooth the informative coverage path.
return τ.

Algorithm 2 ICPS_MDP(τ, ϕ)

Require: An informative coverage path τ and a sensor function ϕ.
Ensure: An informative coverage path τ.

while $||u|| > \epsilon$ **do**
  for all $p_i \in \tau$ **do**
    Compute Voronoi-like partition V_i using k-nearest-neighbor (KNN) algorithm for p_i.
    Run PBVI to generate (Φ_{p_i}, π_{p_i}).
    Integrate the system dynamics of $\dot{p}_i = u_i = f(p_i)$ using (4) and the Euler approximation [13], i.e. $p_{k+1,i} = p_{k,i} + u_{k,i} \cdot \Delta t_k$
  end for
  $k \leftarrow k + 1$
end while
return τ

Algorithm 3 FINDCOLLISIONFREEWAYPOINTS$(\tau, \mathcal{O})$

Require: An informative coverage path τ, and an obstacle region $\mathcal{O} \subset X_d$.
Ensure: An informative coverage path τ.

Compute $\mathcal{O}_D$ by dilating $\mathcal{O}$ using the convolution mask $\mathbf{1}_{3\times 3}$, or a 3×3 matrix of ones.
for all $p_i \in \tau$ **do**
  if $p_i \in \mathcal{O}_D$ **then**
    Eliminate p_i if the local direction of τ does not change more than 5 degrees, i.e., if $|\theta_i| < 5^\circ$. Here $\cos\theta_i = \frac{\vec{p}_{i+1,i-1} \cdot \vec{p}_{i,i-1}}{||\vec{p}_{i+1,i-1}|| \, ||\vec{p}_{i,i-1}||}$ and $\vec{p}_{i,j} = p_i - p_j$ is the vector difference of p_j and p_i.
  else
    if $\exists$ an adjacent waypoint $p_j \notin \mathcal{O}_D$ **then**
      Apply a variant of the bisection numerical method to push p_i along the line segment $\overline{p_i, p_j}$ the minimum distance until $p_i \notin \mathcal{O}_D$, or the length of the segment being divided decreases below a user-specified threshold.
    else
      Search for a waypoint $p_j \notin \mathcal{O}_D$ along the line segments $\overline{p_i, p_{i-1}}$ and $\overline{p_i, p_{i+1}}$ using the Van Der Corput sequence [13] until the first $p_j \notin \mathcal{O}_D$ is found or the mesh of points searched decreases below a user-specified threshold.
    end if
    if $p_j \notin \mathcal{O}_D$ is not found **then**
      Eliminate p_i.
    end if
  end if
end for
return τ.

Algorithm 4 ELIMINATEWAYPOINTS(τ, ϕ)

Require: An informative coverage path τ and a sensor function ϕ.
Ensure: An informative coverage path τ.

Calculate mass and area of each waypoint's $p_i \in \tau$ Voronoi partition [13] with respect to the sensor function ϕ.
Select $q, r \in \{0, 1, 2, \ldots, |\tau| - 1\}$.
Sort the waypoints of the informative coverage path τ in ascending order with respect to their associated mass and area.
Keep any waypoint that exceeds the q^{th} order mass statistic and the r^{th} order area statistic.
return τ.

Coverage plan τ is executed by starting at one waypoint p_{i-1}, and using the feedback plan π_{p_i}. Once the vehicle reaches the goal state of p_i, then the planner follows the next waypoint feedback plan $\pi_{p_{i+1}}$. It is important to note that for Alg. 4, waypoint statistics are correlated with their neighbors, so eliminating too many waypoints at once may cause entire sections of the path to disappear. We also typically assume that $q = r$.

D. Calculating Plan Execution Success Probabilities

Calculating the probability of collision while executing coverage plan τ is a nontrivial task, because all possible trajectories leaving from one waypoint p_i to get to the next waypoint p_{i+1} must be considered. Given the MDP and its solution, we define a Markov Chain MC $= (X_d, T)$ on X_d with transition probabilities given by $\Theta(x, \pi(x))$, defining state transition matrix $T = [T_{i,j}]$, where $T_{i,j} = \Pr[x_{k+1} = j | x_k = i]$. We further assume that Θ satisfies the property that if $x_i \in \mathcal{O} \cup X_{goal}$, then $T_{i,j} = \delta_{ij}$, where δ_{ij} is the Kronecker delta function. In other words, vehicles that enter goal or obstacle locations are forever trapped. The stationary distribution Y of T (satisfying $Y = YT$) can be used to calculate the probability of success of a coverage plan being executed. Y is a probability mass function in the belief space of X_d, i.e., $Y^{(i)} = \Pr[X = x_i] \; \forall x_i \in X_d$. The event $\{x_i \in \mathcal{O}\}$ denotes that a collision occurred, while the event $\{x_i \in X_{goal}\}$ denotes a successful arrival at the goal waypoint. Given Y, the probabilities of these events can be calculated. We approximate Y numerically by executing the fixed point method $Y_{k+1} = Y_k T$ until it converges (T is sparse). Note that Y exists but is not necessarily unique. However, we are most concerned with the limiting distribution Y when Y_0 is a delta distribution centered at the previous waypoint. $P_{succ}^{(i)}$ is the probability of reaching p_{i+1} from p_i without collision and P_{succ} is the probability of successfully executing the coverage plan τ, i.e., $P_{succ} = \prod_{i=1}^{N} P_{succ}^{(i)}$.

III. EXPERIMENTAL SETUP

A set of three distinct environments were created: "Gaussian bump grid" $\mathcal{E}_1$ (see Fig. 2 and Fig. 4), "trail fork" $\mathcal{E}_2$ (see Fig. 5), and "barricaded hot spots" $\mathcal{E}_3$ (see Fig. 5). These environments were augmented by inclusion of a wind gust occupancy grid $\tilde{\mathcal{E}} = \mathcal{E} \times \mathbf{m} = (\phi, \mathcal{O}, \mathbf{m})$ [18], where $\mathbf{m}$ is a collection of Bernoulli random variables that define the probability of a gust of wind occurring from the north (in the $-x$ direction) in a particular cell. In other words, $x_i \in X_d$ is assigned a particular $m_i \in \mathbf{m}$. We assume that $\Pr[\theta_k | x_k, u_k, m_k]$ can be more readily defined, and m_k can be marginalized out yielding $\Theta(x_k, u_k)$ using the law of total probability (the distribution of $\mathbf{m}$ is given). This allows us to localize the locations of high probability wind gusts to *particular regions in the environment*, highlighting that this induces local effects in the informative plan. For each environment, $\mathcal{E}_a$, $a \in \{1, 2, 3\}$, two different occupancy grids $\mathbf{m}_0$ and $\mathbf{m}_a$ were used, yielding a total of 6 simulations using augmented environments. In other words, the 6 environments tested were $\{(\mathcal{E}_a, \mathbf{m}_0)\} \cup \{(\mathcal{E}_a, \mathbf{m}_a)\}$. Here, $\mathbf{m}_0$ is a uniform low probability wind gust occupancy grid, where $\Pr[m_i = 1] = 0.1 \; \forall i$, and $\mathbf{m}_a$ was tailored for its respective environment $\mathcal{E}_a$ to have high probability wind gust regions properly located to attempt to blow the vehicle into a given obstacle.

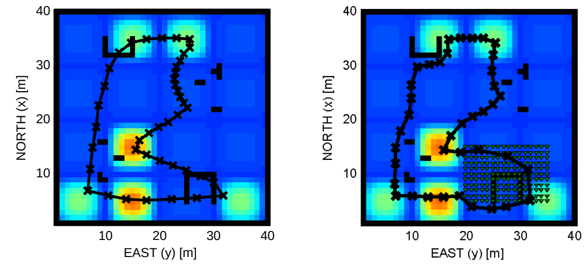
To ease visualization purposes, $\Pr[m_i = 1]$ is constrained to be either 0.1 or 1. The locations where $\Pr[m_i = 1] = 1$ are denoted by a green downward pointing triangle to indicate the direction the wind is blowing.

For all simulation runs, the descent direction $\hat{h} = \hat{h}_{mode}$ was selected (5). Apart from the environment, simulation parameters for all runs were identical, including the initial waypoint selection in Fig. 2(a). Obstacles were constructed such that even after obstacle dilation, the free space $X_{D,free} = X_d \setminus \mathcal{O}_D$ remains fully connected.

IV. SIMULATION RESULTS

After executing the proposed algorithm, the resulting locations of the waypoints for the environments "Gaussian bump grid" is shown in Fig. 4, while both "trail fork" and "barricaded hot spots" are in Fig. 5. The coordinate frame follows the North-East-Down (NED) convention.

One important performance metric previously unspecified in the informative path planning literature is the probability of path execution success (conversely, probability of obstacle collision during path execution) while executing the feedback plans to traverse between waypoints. For each simulation, we compare the vehicle's probability of success while executing the specified coverage plan. We compare the waypoints selected using the original algorithm ICPS proposed in [5] to the ICPS_MDP algorithm. Note that ICPS is a part of ICPS_MDP, so the coverage plans generated by the original algorithm are shown in Figs. 4(a), 5(a), and 5(b). Since the original ICPS algorithm has no knowledge of obstacles in the environment, waypoints inside obstacles are eliminated, and MDP-based feedback plans are generated to steer the vehicle around obstacles when computing plan execution success probabilities.



(a) Waypoint locations after executing ICPS. (b) Wind blowing from north to south.

Fig. 4. "Gaussian bump grid" environment simulation results. This is the same environment as shown in Fig. 2, contrasting the ICPS and ICPS_MDP simulation results. Notice here that the north wind present in the bottom right in 4(b) causes the vehicle to move waypoints upwind to reduce the probability of collision.

To test the robustness of the coverage plan, we also incorporate uncertainty into the planner's estimate of Θ so there is a mismatch between the environment and the planner's model of the environment. We define a new wind occupancy grid whose parameters are a mixture distribution of the true occupancy grid, and uniform noise: $\Pr[\hat{m}_i = 1] = \alpha Pr[m_i = 1] + (1-\alpha)\zeta_i$, where ζ_i is a continuous R.V. s.t. $\zeta_i \sim Unif(0, 1)$ and we set $\alpha = 0.5$.

Twenty wind occupancy grids $\hat{\mathbf{m}}$ for each simulation result were drawn from ζ_i, and success probability statistics were calculated in Table I.

Trial	wind	1) Gauss.	2) Trail	3) Hot
ICPS [5] [$P_{succ.}$]	high	88%	89%	95%
ICPS [5] [σ]	high	0.027	0.019	0.016
ICPS_MDP [$P_{succ.}$]	high	99.9%	94%	98%
ICPS_MDP [σ]	high	0.0013	0.019	0.0043
ICP [5] [$P_{succ.}$]	low	∼100%	∼100%	99.6%
ICP [5] [σ]	low	10^{-9}	10^{-9}	0.014
ICPS_MDP [$P_{succ.}$]	low	∼100%	∼100%	∼100%
ICPS_MDP [σ]	low	10^{-15}	10^{-9}	10^{-13}

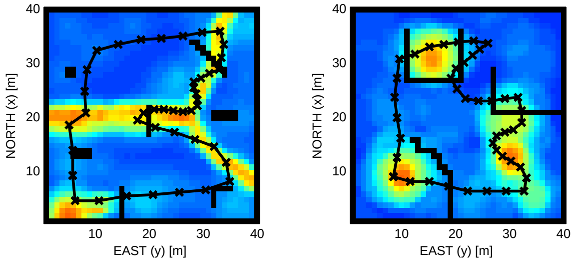
TABLE I

PLAN EXECUTION SUCCESS PROBABILITIES FOR THE ENVIRONMENTS.

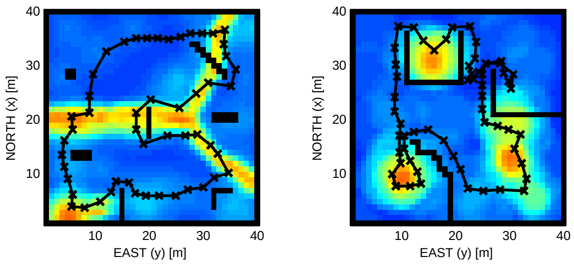
V. CONCLUSIONS

In this paper, we have introduced a planning algorithm for computation of informative coverage plans for persistent monitoring tasks with static obstacles. The algorithm morphs an initial informative coverage path towards regions with high information value, while avoiding obstacles and explicitly considering environmental effects that cause uncertainty in a vehicle's motion. Simulation results show that the proposed algorithm is robust to motion uncertainties and hence reduces the probability of collision with obstacles in the environment.

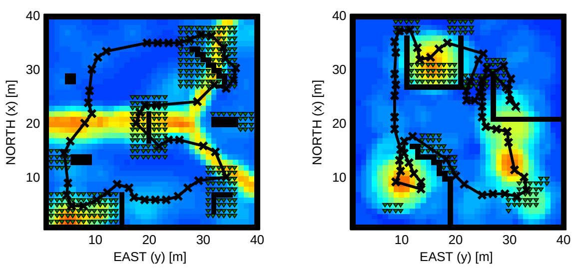
In future work, positioning uncertainty stemming from sensor inaccuracies can be modeled by using a Partially Observable MDP [18]. For this work, the sensor function has been given *a priori* as a part of the environment. The



(a) "Trail fork": waypoint locations after executing ICPS. (b) "Hot spots": waypoint locations after executing ICPS.



(c) "Trail fork": low occurrence of wind. (d) "Hot spots": low occurrence of wind.



(e) "Trail fork": wind blowing from north to south. (f) "Hot spots": wind blowing from north to south.

Fig. 5. For "Trail fork" simulation results: the waypoints near the wind field in the top right corner seems to work correctly and are pushed to the wind gust boundary. Otherwise, wind effects seem to be negligible in this environment. However, the planner does not seem to have larger clearances for the lower obstacles. For "Barricaded hot spots" simulation results: notice what appears to be an extraneous traversal into the region in the top right of both figures. This could be a consequence of the relative weighting W_s, W_g of costs in (3). A further improvement of waypoint elimination heuristics could reduce the number of waypoints in the bottleneck.

characterization of relevant sensor functions using onboard, possibly noisy sensors selected for a particular mission deserves further study.

Acknowledgments

This work was performed in part at the Naval Research Laboratory and was funded by the US Department of Defense, Office of Naval Research, under grant number N0001413WX21045, Mobile Autonomous Navy Teams for Information Surveillance and Search (MANTISS). S.K. Gupta's participation in this research was supported by National Science Foundation's Independent Research and Development program. The authors would like to thank Keith Sullivan and Thomas Apker at NRL for their insights, the rest of the Laboratory for Autonomous Systems Research at NRL, and the rest of the Simulation-Based System Design Laboratory at UMD.

References

[1] H. Choset, "Coverage for Robotics - a Survey of Recent Results," *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1-4, pp. 113–126, 2001.

[2] J. Binney and G. S. Sukhatme, "Branch and Bound for Informative Path Planning," in *IEEE International Conference on Robotics and Automation (ICRA), 2012*. IEEE, Saint Paul, MN, May 14-18, 2012, pp. 2147–2154.

[3] D. Levine, B. Luders, and J. P. How, "Information-Rich Path Planning with General Constraints Using Rapidly-Exploring Random Trees," in *AIAA Infotech Aerospace Conference, Atlanta, GA*, 2010.

[4] S. Smith, M. Schwager, and D. Rus, "Persistent Robotic Tasks: Monitoring and Sweeping in Changing Environments," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 410–426, 2012.

[5] D. E. Soltero, M. Schwager, and D. Rus, "Generating Informative Paths for Persistent Sensing in Unknown Environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012*. IEEE, Vilamoura, Algarve, October 7-12, 2012, pp. 2172–2179.

[6] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage Control for Mobile Sensing Networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[7] R. Graham and J. Cortés, "Adaptive Information Collection by Robotic Sensor Networks for Spatial Estimation," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1404–1419, 2012.

[8] B. J. Julian, M. Angermann, M. Schwager, and D. Rus, "Distributed Robotic Sensor Networks: An Information Theoretic Approach," *The International Journal of Robotics Research*, vol. 31, no. 10, pp. 1134–1154, 2012.

[9] S. Carpin, D. Burch, and T. H. Chung, "Searching for Multiple Targets Using Probabilistic Quadtrees," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2011*. IEEE, San Francisco, CA, September, 25-30, 2011, pp. 4536–4543.

[10] G. Hollinger and S. Singh, "Multi-Robot Coordination with Periodic Connectivity," in *IEEE International Conference on Robotics and Automation (ICRA), 2010*. IEEE, Anchorage, AK, May 3-7, 2010, pp. 4457–4462.

[11] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient Informative Sensing Using Multiple Robots," *Journal of Artificial Intelligence Research*, vol. 34, no. 2, p. 707, 2009.

[12] C. T. Cunningham and R. S. Roberts, "An Adaptive Path Planning Algorithm for Cooperating Unmanned Air Vehicles," in *IEEE International Conference on Robotics and Automation (ICRA), 2001*, vol. 4. IEEE, 2001, pp. 3981–3986.

[13] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[14] A. Thakur, P. Svec, and S. K. Gupta, "GPU Based Generation of State Transition Models Using Simulations for Unmanned Sea Surface Vehicle Trajectory Planning," *Robotics and Autonomous Systems*, Vol. 60, no. 12, pp. 1457–1471, 2012.

[15] P. Švec, B. C. Shah, I. R. Bertaska, J. Alvarez, A. J. Sinisterra, K. von Ellenrieder, M. Dhanak, and S. K. Gupta, "Dynamics-Aware Target Following for an Autonomous Surface Vehicle Operating Under COLREGS in Civilian Traffic," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'13)*, Tokyo, Japan, November 3-7, 2013, pp. 3871–3878.

[16] S. Salapaka, A. Khalak, and M. Dahleh, "Constraints on Locational Optimization Problems," in *IEEE Conference on Decision and Control (CDC), 2003*, vol. 2. IEEE, Maui, HI, December 9-12, 2003, pp. 1741–1746.

[17] L. C. Pimenta, M. Schwager, Q. Lindsey, V. Kumar, D. Rus, R. C. Mesquita, and G. A. Pereira, "Simultaneous Coverage and Tracking (SCAT) of Moving Targets with Robot Networks," in *Algorithmic Foundation of Robotics VIII*. Springer, 2009, pp. 85–99.

[18] S. Thrun, W. Burgard, D. Fox *et al.*, *Probabilistic Robotics*. MIT Press Cambridge, 2005, vol. 1.